

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A system for providing a floating point division, comprising:

an analyzer circuit configured to determine a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

a results circuit coupled to the analyzer circuit and configured to assert a resulting floating point operand containing:

the result of the division of the first floating point operand by the second floating point operand in a first portion of the resulting floating point operand;
and

a resulting status embedded within a second portion of the resulting floating point operand, the second portion being distinct from the first portion.

2. (Original) The system for providing a floating point division of claim 1, wherein the analyzer circuit further comprises:

a first operand buffer configured to store the first floating point operand;

a second operand buffer configured to store the second floating point operand;

a first operand analysis circuit coupled to the first operand buffer, the first operand analysis circuit configured to generate a first characteristic signal having information relating to the first status; and

a second operand analysis circuit coupled to the second operand buffer, the second operand analysis circuit configured to generate a second characteristic signal having information relating to the second status.

3. (Original) The system for providing a floating point division of claim 2, wherein the first status and the second status are determined without regard to memory storage external to the first operand buffer and the second operand buffer.

4. (Original) The system for providing a floating point division of claim 3, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

5. (Original) The system for providing a floating point division of claim 1, wherein the results circuit further comprises:

a divider circuit coupled to the analyzer circuit, the divider circuit configured to produce the result of the division of the first floating point operand by the second floating point operand;

a divider logic circuit coupled to the analyzer circuit and configured to produce the resulting status based upon the first status and the second status; and

a result assembler coupled to the divider circuit and the divider logic circuit, the result assembler configured to assert the resulting floating point operand and embed the resulting status within the resulting floating point operand.

6. (Original) The system for providing a floating point division of claim 5, wherein the divider logic circuit is organized according to the structure of a decision table.

7. (Original) The system for providing a floating point division of claim 1, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

8. (Original) The system for providing a floating point division of claim 7, wherein the overflow status represents one in a group of a +ON status and a -ON status.

9. (Original) The system for providing a floating point division of claim 8, wherein the overflow status is represented as a predetermined non-infinity numerical value.

10. (Original) The system for providing a floating point division of claim 7, wherein the underflow status represents one in a group of a +UN status and a -UN status.

11. (Original) The system for providing a floating point divider of claim 10,
wherein the underflow status is represented as a predetermined non-zero numerical
value.

12. (Original) The system for providing a floating point division of claim 7,
wherein the invalid status represents a not-a-number (NaN) status due to an invalid
operation.

13. (Original) The system for providing a floating point division of claim 7,
wherein the infinity status represents one in a group of a positive infinity status and a
negative infinity status.

14. (Currently Amended) A method for providing a floating point division,
comprising:

determining, using a divider unit, a first status of a first floating point
operand and a second status of a second floating point operand based upon data within
the first floating point operand and data within the second floating point operand
respectively; and

asserting a resulting floating point operand containing:

the result of the division of the first floating point operand by the
second floating point operand in a first portion of the resulting floating point operand;
and

floating point operand, the second portion being distinct from the first portion.

15. (Original) The method for providing a floating point division of claim 14, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer;
storing the second floating point operand in a second operand buffer;
generating a first characteristic signal representative of the first status; and
generating a second characteristic signal representative of the second

status.

16. (Original) The method for providing a floating point division of claim 15, wherein the first characteristic signal and the second characteristic signal are generated without regard to memory storage external to the first operand buffer and the second operand buffer.

17. (Original) The method for providing a floating point division of claim 16, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

18. (Original) The method for providing a floating point division of claim 14, wherein the asserting stage further comprises:

producing the result of the division of the first floating point operand by the second floating point operand; and
asserting the resulting floating point operand having the resulting status embedded within the resulting floating point operand.

19. (Original) The method for providing a floating point division of claim 14, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

20. (Original) The method for providing a floating point division of claim 19, wherein the overflow status represents one in a group of a +OV status and a -OV status.

21. (Original) The method for providing a floating point division of claim 20, wherein the overflow status is represented as a predetermined non-infinity numerical value.

22. (Original) The method for providing a floating point division of claim 19, wherein the underflow status represents one in a group of a +UN status and a -UN status.

23. (Original) The method for providing a floating point division of claim 22, wherein the underflow status is represented as a predetermined non-zero numerical value.

24. (Original) The method for providing a floating point division of claim 19, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

25. (Original) The method for providing a floating point division of claim 19, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

26. (Currently Amended) A computer-readable medium on which is stored a set of instructions for providing a floating point division, which when executed perform stages comprising:

determining a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

asserting a resulting floating point operand containing;

the result of the division of the first floating point operand by the second floating point operand in a first portion of the resulting floating point operand;
and

floating point operand, the second portion being distinct from the first portion.

27. (Original) The computer-readable medium of claim 26, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer;
storing the second floating point operand in a second operand buffer;
generating a first characteristic signal representative of the first status; and
generating a second characteristic signal representative of the second status.

28. (Original) The computer-readable medium of claim 27, wherein the first characteristic signal and the second characteristic signal are generated without regard to memory storage external to the first operand buffer and the second operand buffer.

29. (Original) The computer-readable medium of claim 28, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

30. (Original) The computer-readable medium of claim 26, wherein the asserting stage further comprises:

producing the result of the division of the first floating point operand by the second floating point operand; and

asserting the resulting floating point operand having the resulting status
embedded within the resulting floating point operand.

31. (Original) The computer-readable medium of claim 26, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

32. (Original) The computer-readable medium of claim 31, wherein the overflow status represents one in a group of a +OV status and a -OV status.

33. (Original) The computer-readable medium of claim 32, wherein the overflow status is represented as a predetermined non-infinity numerical value.

34. (Original) The computer-readable medium of claim 31, wherein the underflow status represents one in a group of a +UN status and a -UN status.

35. (Original) The computer-readable medium of claim 34, wherein the underflow status is represented as a predetermined non-zero numerical value.

36. (Original) The computer-readable medium of claim 31, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

U.S. Application No. 10/035,647

Appeal 2007-2270

Attorney Docket No. 06502.0379-00

37. (Original) The computer-readable medium of claim 31, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.